IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 June 2023, accepted 19 June 2023, date of publication 26 June 2023, date of current version 2 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3289404

SURVEY

Experimental and Theoretical Study for the Popular Shilling Attacks Detection Methods in Collaborative Recommender System

REDA A. ZAYED[1,2], **LAMIAA FATTOUH IBRAHIM**[1], **HESHAM A. HEFNY**[1], **HESHAM A. SALMAN**[2], **AND ABDULAZIZ ALMOHIMEED**[3]

[1]Faculty of Graduate Studies for Statistical Research, Cairo University, Giza 12613, Egypt
[2]College of Informatics, Midocean University, Moroni 8722, Comoros
[3]Computer Science Department, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 13318, Saudi Arabia

Corresponding author: Lamiaa Fattouh Ibrahim (lfattouh@cu.edu.eg)

This work was supported in part by Cairo University, Egypt; and in part by Midocean University, Comoros.

ABSTRACT The stability and reliability of filtration and recommender systems are crucial for continuous operation. The presence of fake profiles, known as ''shilling attacks,'' can undermine the reliability of these systems. Therefore, it is important to detect and classify these attacks. Numerous techniques for detecting shilling attacks have been proposed, including supervised, semi-supervised, unsupervised, Deep Learning, and hyper deep learning methods. These techniques utilize well-known shilling attack models to target collaborative recommender systems. While previous research has focused on evaluating shilling attack strategies from a global perspective, considering factors such as attack size and attacker's knowledge, there is a lack of comparative studies on the various existing and commonly used attack detection methods. This paper aims to fill this gap by providing a comprehensive survey of shilling attack models, detection attributes, and detection algorithms. Furthermore, we explore the traits of injected profiles that are exploited by detection algorithms, which has not been thoroughly investigated in prior works. We also conduct experimental studies on popular attack detection methods. Our experimental results reveal that hybrid deep learning algorithms exhibit the highest performance in shilling detection, followed by supervised learning algorithms and semi-supervised learning algorithms. In contrast, the unsupervised technique performs poorly. The deep learning-based Shilling Attack Detection demonstrates accuracy and quality in accurately identifying a variety of mixed attacks. This study provides valuable insights into shilling attack models, detection attributes, and detection algorithms. Our findings highlight the superior performance of hybrid deep learning algorithms in shilling detection, as well as the limitations of unsupervised techniques. Deep learning-based Shilling Attack Detection showcases its effectiveness and accuracy in identifying various types of attacks.

INDEX TERMS Shilling attack, profile injection, collaborative recommender system, machine learning.

I. INTRODUCTION

These days whether you see or observe a video on YouTube, a motion movie on Netflix or an item on Amazon, you are reaching suggestions for more other things to see, like or purchase. You would like to thank the advent of artificial intelligence, machine learning, and recommender frameworks for this errand. Recommender frameworks give personalized data by learning the user profiles and their activities. Similar machine learning algorithms, a recommender framework makes a forecast based on user activity [1], which was created to predict user preference for a collection of items based on previous experience. The information is filtered through collaborative filtering, which utilizes user interactions and data from other users. It is assumed that people who agree in their assessment of specific objects will probably happen again in the future. Collaborative recommender systems use similarity index-based method. In the neighborhood-based

The associate editor coordinating the review of this manuscript and approving it for publication was Mouloud Denai.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

approach, many users are selected depending on how close they are to the active user. Calculating the weighted average of the ratings of the selected users yields an inference for the active user. Collaborative filtering systems focus on the relationship between users U and items I (U × I). The similarity of items is established by the similarity of their ratings by users who have rated both items. Recommender systems (RS) are one of the most important components in providing decision making predictions, recommender systems have efficient methods and procedures for dealing with massive amounts of data [2]. Individuals can utilize recommendation systems to make decisions and can find the most relevant ideas from many options. Users of recommender systems can provide ratings for items and products. The rating operation must be protected from manipulation by recommender systems. Collaborative recommenders are among the most active and effective recommendation systems currently available, providing excellent suggestions and recommendations [3]. The techniques and classification of recommender systems are classified as content based, collaborative and hybrid filtering as shown in figure 1.

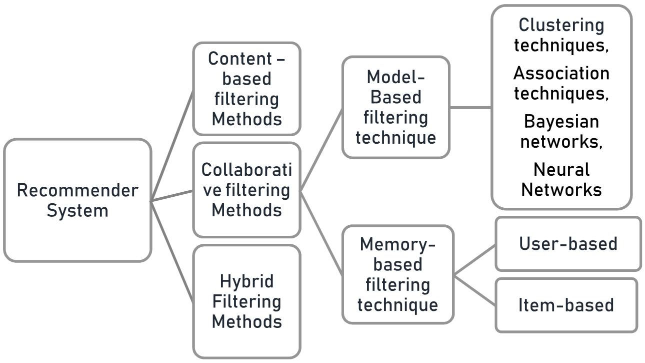

FIGURE 1. Approaches to recommender systems: An overview [1].

The similarity and relationship between user preferences are used in collaborative filtering (CF). Consequently, user correlations are the most important aspect in the suggestion process [3]. Even though employing user associations allows for the creation of acceptable neighborhoods, it creates a vulnerability in collaborative filtering algorithms. Attackers attempting to manipulate the results of a recommender system by injecting and pushing fake profiles into the target system. The injected profiles are well formatted to have similar genuine and brilliant active profiles. "Shilling attacks" refer to the work of creating bogus profiles and injecting them into the system. Recent research has found that collaborative filtering approaches are vulnerable to these types of attacks [4]. The process of designing the fake profiles and injecting them into the system is called "shilling attacks" and the recent studies mention that collaborative recommender applications are weak and vulnerable to these kinds of attacks. Detecting shilling attacks is typically viewed as a binary classification problem, in which the classification results for each profile might be an active genius user or an abnormal(fake) user "attacker" [2]. The detection feature techniques introduced use machine learning to detect and classify the attackers from genius active profiles. In this study, we conducted an experimental survey for popular detection methods that pick-up well-known detectors for shilling attacks on collaborative recommender systems. This study aims to provide a comprehensive survey of different attack models and detection methods for shilling attacks on recommender systems to build road maps to assess of the current state of research on recommender system attacks and detection techniques.

II. BACKGROUND

The attack is carried out in a recommender system by adding shilling profiles to cause bias on target items [5], the purpose of which attacks is to artificially change the rating of individual items by users to increase their sales. When creating malicious users, almost all the attack models employ the same attack profile.

A. ATTACK PROFILE

The shilling attack profile structure contains four sets of items: I_s, I_F, I_t and $I_\emptyset$ as shown in figure 2 [6].

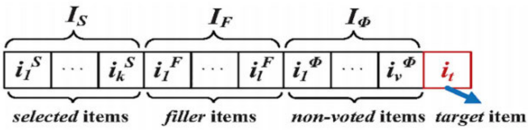

FIGURE 2. The shilling attack profile structure in its most basic version [1].

B. ATTACKS MODELS

Attacks classified into twon main types push and Nuke attacks. This classification is based on the attack purpose, Nuke Attack to demotion of an item, minimum rating given to the target item. Push Attacks to promote a specific item where the maximum ratings are given to the target item [7]. To carry out the requisite shilling attack, the attacker must first gather and obtain knowledge of the target recommender system, after which the attacker can begin the attack. Not only the standard deviation and mean rating for each product or service, but also the ratings and user distribution in the user-item matrix be included in this information. Collecting knowledge and information about the target domain is an important task that must be completed to select appropriate items and ratings for use in creating attack profiles. The famous fake profiles "random, bandwagon, average, segment, reverse bandwagon, and "love/hate" are discussed [2]. The following two points clarify the main type of attacks and classified them to standard and obfuscated attacks.

C. STANDARD ATTACKS

These attack models do not try to go unnoticed in a recommender system to avoid detection. Many detection techniques are more likely to identify shilling attack characteristics incorporated with these attacks.

1) RANDOM ATTACK

The basic and simplest attack is a random attack, which is less effective for all. It creates and builds profiles in which all items rated randomly, except for the target item.

2) AVERAGE ATTACK MODEL

The profiles created by the average attack share the tendencies of the systems users. This is possible by drawing harmful profiles ratings from the rating statistical distribution of that correlates with every item. These attack types force the attacker to collect slight knowledge and detailed information about the item dataset on which the recommender system creates the suggestion and recommendations.

3) BANDWAGON ATTACK MODEL [8]

The attacker determines the item's popularity independently from the system dataset, which is often known as a "popular attack." The attacker restricts the known items, then builds a relationship between the attacked things and a subset of them. This approach delivers many of the advantages of the average attack without detailed knowledge and information about the recommender dataset.

4) THE REVERSE BANDWAGON ATTACK

is the exact opposite of a bandwagon attack which nukes the target product by assigning low ratings to items with a many bad reviews and the lowest rating to the target item. Like the bandwagon assault, it is likewise a low-knowledge attack. The reverse bandwagon assault is marginally more efficient than the bandwagon approach.

5) SEGMENTED ATTACK [8]

This requires less knowledge about the recommender system. The basic concept behind this attack is to popularize the target items among a group of targeted users. For example, the author of a romantic novel wants his novel to be recommended to the readers who are lovers of "The Notebook" (another romantic novel), not to those ones who like comics. In the segmented attack in which the attacker concentrates on a set of items with similar content that have high visibility. Almost all the attack approaches are successful with user-based collaborative filtering algorithms, but they are not as effective as item-based algorithms. The reverse bandwagon attack is a lone exception, as it is exclusively meant to nuke rather than propel objects. Item-based collaborative filtering algorithms are often more difficult to assault upon. One explanation for this is that an item-based algorithm uses the target user's ratings to launch attacks. An intended user is always a genuine individual. Evidently, false profile injection cannot be used to influence a real user stated ratings.

6) PROBE ATTACK [8]

This is not a general-purpose attack that can be used on any system. Some recommender systems project a predicted rating score for each item. This information is used by the attacker to rank the products, allowing them to be compared with other users. Some initial data receive legitimate ratings from the attacker using the recommender system, and then the attacker generates a list of rated objects based on the items suggested by the recommender. This method keeps the attack profiles near its neighbors. It also allows the attacker to give a better understanding of the system.

7) LOVE/HATE ATTACK [9]

The Love/Hate Attack is a powerful nuclear attack. The attacker randomly selects filler objects and assigns the highest ratings to them while assigning the lowest ratings to the target item. Despite its simplicity, this strategy has a remarkable level of efficacy. It was built primarily for nuclear strikes; however, by changing the ratings, it may also be utilized for a push attack. A Nuclear attack is more effective than a push attack.

D. OBFUSCATED ATTACKS

Attackers strive to conceal their attack signature in order to avoid detection using detection algorithms [10]. To create obfuscation, several models have made minor adjustments to typical attack tactics.

1) NOISE INJECTION [10]

For a subset of injected profiles, each rating is given a constant-multiplied Gaussian distributed random number a part of the Noise Injection. The degree of obfuscation was determined by the multiplying constant. It efficiently conceals its signature using all of the conventional attack methods. Because noise injection affects the rating method, there is a small but noticeable decline in the attack efficiency.

2) USER SHIFTING [10]

is an obfuscation technique in which a section of each inserted profiles rated item is changed. To diminish the similarity across attack profiles, the ratings of this collection of elements were boosted or lowered. Distinct subsets of rated items have their ratings adjusted for the different groups of injected profiles.

3) TARGET SHIFTING [10]

This push attacks, target shifting lowers the target item rating to one level below the greatest attainable. The target rating was adjusted to be higher than the lowest feasible rating in nuclear attacks. This tactic is particularly effective at avoiding detection algorithms that punish consumers who give excessive ratings. If the target item is already popular, target shifting obfuscation makes it more difficult to promote. Other obfuscation strategies should also be used in such instances.

4) AVERAGE OVER POPULAR ITEMS AoP [10]

The AoP attack is "an obfuscated version of the average attack, which chooses the filler items with equal probability from the top x% of the most popular items rather than the entire set of items.

5) MIXED ATTACK [10]

Random, average, bandwagon, and segmented attacks are all used in equal numbers in a mixed attack. To be successful, the detection mechanism must be capable of detecting all the conventional attacks. Several attack strategies have been employed to push and nuke the same target object. This helps in avoiding numerous detection methods.

6) THE POWER ITEM ATTACK [10]

uses power items that are chosen in one of three ways. The collection of objects that can affect the broadest group of items is known as a power item. These factors can change the suggestions for other users. The power items in PIA-AS are the top-N items with the highest aggregate similarity. Only when many consumers rated the same two goods would they be comparable. The criterion for selecting power items in PIA-ID is In-Degree centrality for choosing the power items. The top N of each item is chosen after the similarity of each pair of things is computed using weighted significance. The power items are chosen by PIA-NR based on the number of users.

7) POWER USER ATTACK [10]

Like PIA, the power user attack selects a group of users with the most influence on the largest group of users. The power users in the PUA-AS are the top X users with the highest Aggregate Similarity. The power users in PUA-ID are those who participate in most neighborhoods, as determined by the in-degree centrality notion. Users with the highest ratings on their profiles are known as power users in the PUA-NR.

8) SAShA [10]

is an attack method that improves the performance of traditional CF attack models by extracting semantic information from a knowledge network. A knowledge graph is a logically organized collection of factual, category, and ontological data. The semantic similarity between the target items' knowledge network-generated characteristics and all other things in the system was computed in this assault. This data was then used to create the most efficient set of filler items.

III. DETECTION ALGORITHMS

The task of gathering user behaviors and preferences may expose a flaw in which "malicious" users try to introduce bogus profiles into the system to control and modify the system's outcomes. Because of harmful profiles, the collaborative recommender technique provides shilling attacks with weak and vulnerable predictions and suggestions. Consequently, the detection of shilling profiles is essential for improving the reliability of the recommender system. The detection approach uses attack detection structured as a classification problem to identify and classify this profile and reduce its effects on a collaborative recommender [7]. The most common methods for detecting shilling attacks in collaborative recommender systems are classified into supervised, semi-supervised, statistical(unsupervised) learning, deep learning, and hybrid-based deep Learning Algorithms.

A. SUPERVISED LEARNING ALGORITHMS

To identify attacks, supervised attack detection techniques employ classification models. Multidimensional feature vectors are used to describe the individuals or groups of user profiles. In many circumstances, the same qualities as those used in the unsupervised scenario are used to generate these multidimensional feature vectors. For example, a characteristic of a user profile might be the number of profiles to which the user profile is identical. Multiple attributes pertaining to distinct aspects of different types of attacks can be retrieved. Subsequently, a binary classifier may be trained using known attack profiles labeled as $+1$ and the remaining profiles as -1. A trained classifier is employed to determine whether a profile is legitimate. The number of labelled ground truths available in recommender system datasets is limited.

Zhou et al. [11] introduced a TS-TIA method for detecting suspicious ratings using multidimensional time series. They reorganized all ratings for each item by time series, examined each time series, and looked for suspicious rating segments. In these anomaly rating segments, statistical metrics and target item analysis techniques were used to detect shilling attacks. Their Experiments showed that their proposed method is effective and takes less time to detect items under attack in larger datasets and revealed that the proposed method achieves lower precision in large datasets while using less computing power.

Chirita et al. [12] presented numerous criteria for examining malevolent user rating habits and evaluating their ability to detect shilling attacks. Based on these findings, they developed an approach to defend recommender systems from shilling attacks. The algorithm can be used to track user ratings and remove shilling attacker profiles from the recommendation-making process, ensuring that the suggestions remain of good quality.

Burke et al. [13] established and demonstrated the efficacy of a variety of models for such attacks. This paper explains how to detect and respond to profile injection attacks using a classification method. The efficacy of the most powerful assault models previously investigated was dramatically reduced by using this strategy.

Shilling attacks are characterized in the literature according to their intent and the amount of expertise required to attack a system. Burke et al [14] are categorized as push and nuke attacks based on their intent, and low and high knowledge attacks based on the necessary knowledge. However, attacks may be categorized based on rating categories, applications, and CF algorithms, in addition to intent and knowledge.

Williams et al. [15] proposed a classification method to detect and respond to profile injection attacks. A variety of attributes distinguishing traits found in attack profiles in general are identified, as well as an attribute-generating technique for detecting profiles based on reverse-engineered attack models. The combined advantages of these features and the influence of classifier selection on strengthening the resilience of the recommender system are then demonstrated using three well-known classification methods. Their research showed that when paired with a support vector machine classifier, this strategy considerably minimizes the impact of the most powerful attack models previously investigated.

Zhang et al. [16] described a meta-learning-based detection technique that employs a meta-classifier to combine the outputs of base classifiers and create detection results. The differences between the base classifiers efficiently lowered the correlation of misclassifications and increased the meta-predictive level potential. On different-scale Movie Lens datasets, we undertake comparison trials with a single SVM and the voting-based ensemble technique. The experimental findings suggest that the proposed method may successfully increase accuracy while maintaining a high recall.

Zhou et al [17] proposed a method to detect shilling attacks based on SVM and target item analysis method, their proposed technique contains two phases, the first phase is the Borderline-SMOTE approach is utilized to solve the class imbalance problem in classification; this phase yields a crude detection result; the second phase is a fine-tuning phase in which the target items in the prospective attack profiles set are assessed. To demonstrate the usefulness of the suggested methodology, they tested it on the Movie Lens 100 K Dataset and compared its performance to that of previous shilling detection approaches.

To increase the detection performance, Yang et al. [18] extracted well-designed characteristics from user profiles based on the statistical qualities of various assault models, making difficult detection circumstances more manageable. Then, based on the recovered features, they used a form of AdaBoost called the re-scale AdaBoost (RAdaBoost) as their detection technique, referring to the basic notion of re-scale boosting (RBoosting) and AdaBoost. Finally, a series of tests were carried out using the MovieLens-100K dataset to show that RAdaBoost outperforms competing approaches, such as SVM, kNN, and AdaBoost.

The multiattention-based group recommendation model (MAGRM) proposed by Huang et al [19], the MAGRM, a multiattention-based model, With the help of two closely related modules and training, they produce accurate group recommendations. They trained multi-attention neural networks to recognize the distinctive social characteristics of each group. The second module is then suggested to learn to predict the ratings of groups on items based on the first module. The preference interactions between groups and their members were captured by an attentive neural network. Experimental analyses on two real-world databases revealed that MAGRM performed noticeably better than its competitors.

Using supervised learning techniques, Zayed et al. [20] developed an improved technique for detecting shilling attacks in collaborative recommendation systems. The proposed method results show that when they used the ensemble learning algorithm, the proposed method achieved better accuracy in terms of F1-Measure, Recall, Precision, Macro Avg, Weighted Avg.

B. UNSUPERVISED LEARNING ALGORITHMS

In this situation, ad hoc criteria are utilized to detect fake profiles in unsupervised attack detection algorithms. If a profile (or a major piece) is identical to many other profiles, it is likely that all of them were inserted to create an attack. The primary concept of this class of algorithms is to determine the key properties of attack profiles that differ from legitimate profiles. Unsupervised techniques for detecting fraudulent profiles may be designed using such traits to improve performance, and supervised algorithms require a large amount of labelled data. To train classifiers, most classification-based approaches require a balanced number of attacks and normal profiles. Nearest neighbor classifiers, decision tree methods, rule-based classifiers, Bayes classifiers, Neural Network classifiers, and SVM-based classifiers are some of the techniques used in the early detection algorithms. In the second approach, unsupervised detection algorithms are trained on unlabeled data to solve this problem. Compared to supervised techniques, these methods require significantly less computational work, and the key advantage is that these strategies make online learning easier and improve detection accuracy. The use of unsupervised techniques to detect attack profiles has sparked considerable research attention. Clustering, association rule procedures, and statistical approaches are some of the strategies used.

Bryan et al. [21] and Mehta et al [22] provide a simple unsupervised method for spam detection that takes advantage of the statistical aspects of successful spam profiles to deliver a highly accurate and rapid solution.

Bhaumik et al. [23] generated detection features modelled on fundamental descriptive statistics and showed that unsupervised clustering can be employed successfully for attack detection. They conducted comprehensive experiments and considered their results in several ways. Regardless of the assault technique, their experimental results demonstrated that an attribute-based unsupervised clustering algorithm can detect spam users with high accuracy and misclassify legitimate users less frequently.

Chung et al. [24] recommended Beta-Protection to solve this problem proposed. When testing with data gathered from Movie Lens public websites, it bases its theoretical foundation on beta distribution for quick computation and has a reliable performance.

Blige et al [25] proposed a strategy that is particularly effective in detecting specific assault characteristics such as bandwagon, segment, and average attack, according to their empirical findings, they conduct tests on a benchmark data set to assess the success of attack detection.

Yang et al. [26] described a two-stage unsupervised detection approach to defend against such attacks. Adaptive structure learning, which uses adaptive local and global structure learning, selects more effective characteristics based on the current features of the user and object. Suspected users were identified in the first step using a density-based clustering algorithm based on the given characteristics. The selected item features are then used to locate suspect objects to track down the attackers based on the first-stage findings. Finally, the perpetrators were identified. Extensive studies on the MovieLens-100K dataset show that the proposed method is more successful than competing methods. It is worth noting that fascinating discoveries such as unusual ratings can be made.

Yaojun et al. [10] proposed a multi-view ensemble method to detect shilling attacks in collaborative recommender systems. they introduced a repartition strategy to increase the diversity of views and reduce the influence of feature order, the experimental results on the Netflix and Amazon review datasets showed that the proposed method outperforms benchmark methods in detecting various synthetic and real-world attacks.

Panagiotakis et al. [27] developed a new method to detect malicious ratings that are created in a hurry and injected into recommender systems and a new attribute (RIS) to capture the randomness in item selection of abnormal profiles. they also proposed three different systems to detect abnormal profiles. Their experimental results on the MovieLens and the Small Netflix datasets demonstrate the high performance of the proposed methods as well as the discrimination accuracy of the proposed features.

C. SEMI-SUPERVISED LEARNING ALGORITHMS

Semi-supervised learning is a machine-learning technique that involves training using a small quantity of labelled data and a large amount of unlabeled data. Unsupervised learning (with no labelled training data) and supervised learning are two types of learning (with labelled training data only). This is a unique case of poor supervision [28] when unlabeled data are combined with a modest bit of labelled data, the learning accuracy can be significantly improved.

Wu et al. [29] extracted user attributes from multiple view information, such as rating values, item temporal popularity, and rating timestamps, to describe the shilling profiles, refined the feature set partition approach to link it with the kNN base classifiers and built the Multiview ensemble method to identify distinct shilling profiles based on the suggested features. The experimental findings on the Netflix and Am-azon review datasets show that the suggested features are more effective, and the pro-posed detection approach performs better than the baseline methods.

Zhang et al. [16] proposed a hybrid shilling attack detector. HySAD uses MC-Relief to choose effective detection metrics and Semi-supervised Naive Bayes (SNB lambda) to distinguish Random-Filler and Average-Filler model attackers from regular users. Extensive tests on the Movie Lens and Netflix datasets show that HySAD is successful in detecting hybrid shilling attacks and is resilient to different obfuscation methods. A real-world case study on Amazon.cn product evaluations is also included, demonstrating that HySAD may successfully increase the accuracy of a collaborative filtering-based recommender system, while also providing intriguing prospects for in-depth investigation of attacker activities. Thus, the usefulness of HySAD for real-world applications is justified.

D. DEEP LEARNING ALGORITHMS

Deep learning, also known as deep structured learning, is a machine-learning approach that uses artificial neural networks to learn representations. These three options are unsupervised, semi-supervised, and supervised learning. Deep learning In fields like computer vision, speech recognition, natural language processing, ma-chine translation, bioinformatics, drug design, medical image analysis, climate science, material inspection, and board game programmers, architectures like deep neural networks, deep belief networks, deep reinforcement learning, recurrent neural networks, and convolutional neural networks have been used [30], [31]. These architectures have produced results that are comparable to, and in some cases superior to, traditional methods. Chao et al. [30] proposed a study that showed how to apply CNN-SAD, a unique convolutional neural network-based approach that uses a modified network topology to leverage deep-level characteristics from user rating profiles. CNN-SAD can identify shilling attacks more effectively because the attained deep-level features elaborate user ratings more precisely than artificially produced features. According to the findings of the experiments, the proposed technique accurately detects most obfuscated attacks and outperforms existing state-of-the-art algorithms, which benefits SAN applications and security.

Zhang et al. [32] employed the singular-value decomposition (SVD) approach proposed by Hurley et al. [33].

Mehta et al. [34] constructed both supervised and unsupervised detectors using the Neyman-Pearson theory. Unsupervised shilling attack detection using principal component analysis (PCA) proposed in [35]. Profile injection attacks have also been detected using statistical techniques [36]. The other method is a semi-supervised detection approach, in which researchers such as Blige et al. [25] used both unlabeled and labelled user profiles. Using both types of data, a new semi-supervised algorithm called semi-SAD shilling-assault detection was proposed. Hyatt used MC-Relief to identify effective detection metrics and semi-supervised Naive Bayes to precisely distinguish random filler model attackers from average-filler model attackers by Nasir et al. [37]. Sundar et al. [9] conducted a complete study of various assault types, detection attributes, and detection techniques, and categorized the inherent characteristics of the inserted profiles employed by the detection algorithms, which had not previously been investigated. They also touch on recent developments in the construction of resilient algorithms that mitigate the effects of shilling attacks, multicriteria system attacks, and intrinsic feedback-based collaborative filtering methods.

Deep-learning networks of many forms, such as GANs and DRLs, have demonstrated great agreement in terms of their success and extensive use with diverse types of data. On the other hand, deep learning algorithms do not model uncertainty in the same manner as Bayesian or probabilistic techniques. Hybrid learning models combine two types of learning to maximize the benefits of each. Bayesian deep learning, Bayesian GANs, and Bayesian conditional GANs are examples of hybrid models [38].

To identify shilling attacks in recommender systems, Vivekanandan et al. [38] proposed a hybrid convolutional neural network (CNN) and long short-term memory (LSTM)-based deep learning model (CNN–LSTM). An altered network architecture is used in this deep learning model to leverage the deep-level properties acquired from user-rated profiles. It addresses the shortcomings of existing shilling attack detection approaches that primarily focus on identifying spam users by artificially inventing characteristics to improve their efficiency and robustness. It is also effective in elucidating deep-level traits for detecting shilling attacks by precisely elaborating user ratings. In comparison to the state-of-the-art algorithms employed for the investigation, the proposed CNN–LSTM technique accurately detected the most obfuscated attacks in the experiments.

To detect shilling attacks efficiently, Ebrahimian et al. [39] presented a hybrid model of two separate neural networks: convolutional and recurrent neural networks. For the qualities generated from the user-rated profiles, the suggested deep learning model employs an altered network architecture. Compared with the state-of-the-art deep learning algorithms used for research, the hybrid model produced better predictions on the Movie-Lens 100 K and Netflix datasets by accurately detecting most of the frequent threats.

Aktukmak et al. [40] proposed using user attributes to detect attacks quickly and accurately in recommender systems. To detect sequential attacks on recommender systems, the proposed technique uses user attributes in a probabilistic model by using the EM technique to optimize the model parameters. The researchers were able to embed mixed-data type user traits as well as ratings into a low-dimensional latent space. To discover persistent outliers, new users are projected into the latent space learned during training using actual user attributes and ratings, and an anomaly is generated in a sequential framework.

To demonstrate the effectiveness of the algorithms, they created a sequential attack scenario on a genuine dataset in which malicious profiles were linked to realistic but random features. Initial tests on the well-known benchmark movie lens dataset show that the proposed approach surpasses the baseline algorithms in terms of detection accuracy and speed, which will be proven in the future with more sophisticated attack models. Zhang et al. [41] introduced GERAI, a GCN-based recommender system that protects consumers from attribute-inference attacks while maintaining usability. GERAI masks user features, including sensitive data, before incorporating differential privacy into the GCN, which effectively bridges user preferences and features for generating secure recommendations, preventing a malicious attacker from estimating and deducing their private attributes from the user interaction history and recommendations. The results show that GERAI is capable of outperforming humans in both recommendation and attribute-protection tasks.

IV. EVALUATION METRICS

To measure and evaluate any proposed model, some evaluation methods are employed, such as the false positive rate, Detection Rate, precision, and recall. "Attacks" is the number of attacks while "Detection" is the number of detected profiles [42].

$$Detection\ Rate = \frac{\#Detection}{\#Attacks} \tag{1}$$

The number of bogus genuine profiles is known as "False Positives," whereas the number of true genius profiles is known as "Actual Profiles."

$$\text{False Positive Rate} = \frac{\text{\#False Positives}}{\text{\#Genuine Profiles}} \tag{2}$$

Many of the proposed methods use precision, recall, and F-measure [43]:

$$Precision = \frac{True\ positive}{True\ positive + False\ positive} \tag{3}$$

$$\text{Recall} = \frac{True\ positive}{True\ positive + False\ Negative} \tag{4}$$

$$\text{F1} - \text{Measure} = \frac{2.Precision.\text{Recall}}{Precision + \text{Recall}} \tag{5}$$

TP denotes the number of shilling profiles correctly classified; FN denotes the number of shilling profiles misclassified as genuine profiles, and FP denotes the number of genuine profiles misclassified as shilling profiles. Because precision and recall are two equally important but mutually contradictory metrics, the F1-measure metric was used to evaluate the overall performance of the detection method. The larger the F1-measure, the better the overall performance.

V. DISCUSSIONS AND EXPERIMENTS

We run comprehensive experiments on various benchmark datasets and detection algorithms, we make the experiments using different types of detecting algorithms. The Supervised Methods include the Degree-SAD detection algorithm [44], Co-Detector, and Bayes Detector [45]. The second methodology is Semi-Supervised Methods called Semi-SAD [46], the third one is Co-Detector [47], the approaches are Unsupervised Methods, namely PCA-Select-Users [43], and the final one is FPA [48]. Our experiments were carried out on two datasets: The Amazon dataset and the Movie Lens dataset. We injected malicious users into the datasets to simulate shilling attacks. Table 1 shows the dataset statistical information.

TABLE 1. Statistical information of data sets.

Data set	*Users #*	*Items #*	*Records #*
Amazon	3921	14711	10368
Movie Lens	1337	2068	37042

All ratings followed a five-star scale, with one representing the lowest rating (disliked) and five representing the highest rating (liked). However, in each movie dataset, the ratings were integers ranging from one to six, with one indicating the lowest rating (disliked) and six indicating the highest (loved). Additionally, we generated five types of push–attacker profiles for injection into the datasets. Table 2 shows the method used to generate shilling attack profiles. For feature extraction, we define the popularity profile and popularity distribution of a user, which are the outputs of the data-preprocessing phase. For detection purposes, it is not necessary to operate on all possible values of popularity distribution [49]. Instead, it is appropriate to consider only a small number of probabilities accumulated over certain intervals. Therefore, we set the range of the popularity distribution into several intervals to obtain the accumulated probability as a feature. The mean popularity of a user (MPU) refers to the mean value of the popularity profile or the mean value of the rated item popularity in a user profile. Figures 3 and 4 show the performances of the detectors. The filler item was set at 10%, attack size was set to 10%, which means the ratio of the injected spammer to active genuine profiles, the target count was set at 20, the target item score was set at 5, and the item had an average score lower than the threshold. Items that have a rating count larger than the minimum count may be chosen as one of the target items for each method in each round of the experiment. The remaining 80% of the data were used for the training phase, and 20% labeled data and 60% unlabeled data were used for training. For PCA-Select-Users, we do not need a training set, but report the results on the common test set. We used the classification report[1] visualizer to display the precision, recall, F1, and support scores of the model.

A. OUR EXPERIMENTAL RESULTS

The following tables from table 3, table 4 and table 5 show the supervised approaches experimental.

Where Support is the number of actual occurrences of the class in the specified dataset.

TABLE 2. Generation methods for five-shilling attack models.

Model	Generation methods
Random Attack	Rate 5 to a target item; give filler items random ratings conforming to a Gaussian distribution with mean 3.6 and standard deviation 1.1
Average Attack	Rate 5 to a target item; the ratings for filler items are distributed around the mean for each item
Sampling Attack	Rate 5 to a target item and its similar items; rate 1 to filler items
Segmented Attack	Rate 5 to a target item and 20 bandwagon movies which are those with the most ratings in the dataset give filler items random ratings conforming to a Gaussian distribution with mean 3.6 and standard deviation 1.1
Bandwagon Attack	Copy existing user profiles including maximum rating records; rate 5 to a target item.

TABLE 3. Degree-SAD evaluation summary with filler size 10%.

Degree-SAD	Precision	Recall	F1-score	Support
Active user	0.7855	0.8164	0.8006	610
Attack user	0.6763	0.6324	0.6536	370
Accuracy			0.7469	980
Macro Avg	0.7309	0.7244	0.7271	980
Weighted Avg	0.7443	0.7469	0.7441	980
Run time:	65.256 s			

TABLE 4. Co-Detector evaluation summary with filler size 10%.

Co-Detector	Precision	Recall	F1-score	Support
Active user	0.8874	0.8164	0.85042	610
Attack user	0.8386	0.6324	0.72104	370
Accuracy			0.78573	980
Macro Avg	0.863	0.7244	0.78764	980
Weighted Avg	0.8676	0.7469	0.80273	980
Run time:	164.299756 s			

TABLE 5. Bayes-Detector evaluation summary with filler size 10%.

Bayes-Detector	Precision	Recall	F1-score	Support
Active user	0.9472	0.941	0.9441	610
Attack user	0.9037	0.9135	0.9026	370
Accuracy			0.923	980
Macro Avg	0.9255	0.9272	0.9223	980
Weighted Avg	0.9308	0.9306	0.9303	980
Run time:	160.03 s			

The experimental result summary of the sim-supervised methods as the following is shown in table 5, table 7 and table 8.

Figures 3 and 4 present a summary of the evaluation matrix of the experimental results.

We compared the effectiveness of supervised, semi-supervised, and unsupervised learning approaches as well as some of the existing techniques for detecting shilling attacks. We performed this using a hybrid deep-learning approach as shown in table 9.

[1] https://www.scikit-yb.org/en/latest/api/classifier/classification_report.html

TABLE 6. FAP evaluation summary with filler size 10%.

FAP	Precision	Recall	F1-score	Support
Active user	0.9485	0.5298	0.6798	610
Attack user	0.0602	0.5114	0.1077	370
Accuracy			0.39375	980
Macro Avg	0.5043	0.5206	0.50032	980
Weighted Avg	0.8991	0.5287	0.66585	980
Run time:	130.021s			

TABLE 7. Semi-SAD evaluation summary table with filler size 10%.

Semi-SAD	Precision	Recall	F1-score	Support
Active user	0.9208	0.941	0.93079	610
Attack user	1	0.9135	0.95479	370
Accuracy			0.9356	980
Macro Avg	0.4604	0.9272	0.415282	980
Weighted Avg	0.8479	0.9306	0.787327	980
Run time:	40.654653 s			

TABLE 8. PCA-Select-Users method evaluation summary table with filler size 10%.

PCA-Select-Users	Precision	Recall	F1-score	Support
Active user	0.9001	0.941	0.92009	610
Attack user	0.006	0.9135	0.011921	370
Accuracy			0.466005	980
Macro Avg	0.4531	0.9272	0.40872	980
Weighted Avg	0.8192	0.9306	0.801353	980
Run time:	2.835881 s			

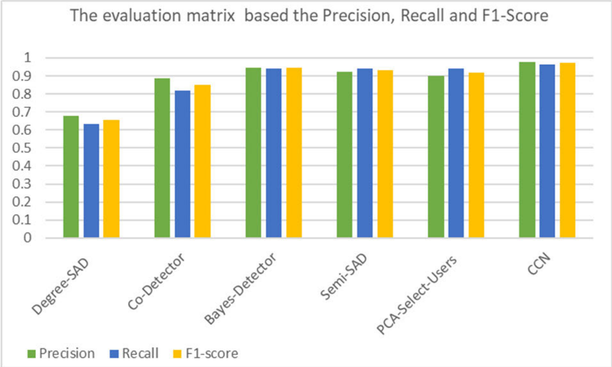

FIGURE 3. The evaluation matrix comparison based the Precision, Recall and F1-Score.

Ebrahimian et al. [48] used the Movie-Lens dataset with the same statistical information as our Movie Lens dataset and compared their 10% attack size results with our experimental results. Figure 5 summarizes the comparison of the detection accuracy between our experiments and some deep learning approaches.

Deep Learning can analyze vast amounts of data and is particularly effective when dealing with massive amounts of data. The relevance of Deep Learning is growing increasingly relevant and common in companies using the CNN-GRU model, achieving an accuracy that exceeds 0.997. Moreover, Bayes Detector, SemiSAD, and PCDSelectUsers achieved an of 0.93.

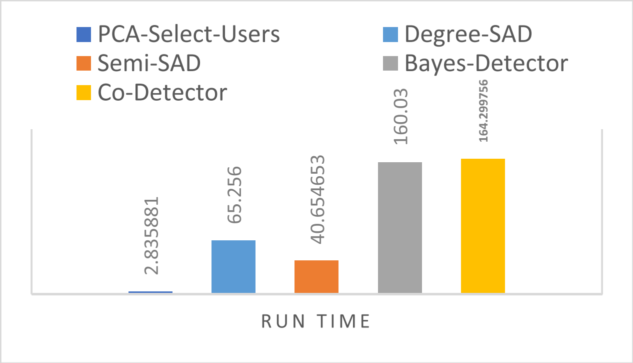

FIGURE 4. The evaluation matrix comparison-based execution run time in second.

TABLE 9. The average summary report for the different techniques.

Detector	Accuracy	Precision	Recall	F1-score	Run time
DegreeSAD	0.7469	0.7855	0.8164	0.7800	65.256 s
CoDetector	0.7469	0.8874	0.8164	0.8504	164.299s
BayesDetector	0.932	0.9472	0.941	0.9440	160.03 s
SemiSAD	0.931	0.9208	0.941	0.9307	40.6546s
PCASelectUsers	0.930	0.9001	0.941	0.9200	2.83588s
FAP	0.5287	0.9485	0.5298	0.6798	130.021s

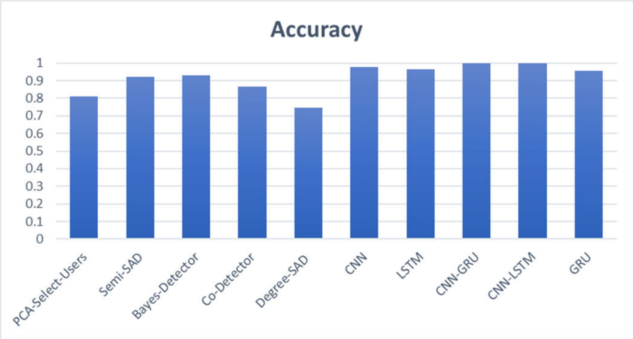

FIGURE 5. Summary of the detection accuracy.

Figure 5 shows the highest accuracy achieved by each model with its corresponding parameters. The CNN-LSTM and CNN-GRU models outperformed single CNN, LSTM, and GRU models.

Our experimental results and survey suggest that deep learning technologies are viable options for more accurate attack detection. Specifically, deep learning techniques followed by supervised learning methods achieve the highest accuracy, but they require more training time. Semi-supervised learning algorithms provide less accuracy, but they are faster to train. Unsupervised learning approaches provides the lowest accuracy.

B. THE THEORETICAL COMPARING WITH OUR EXPERIMENTAL

We compared our experimental results with the original proposed results. Table 10 shows the accuracy of our experimental and other results.

TABLE 10. Comparison of theoretical results accuracy and experimental accuracy for the different techniques.

Detector	Theoretical Results Accuracy	Our Experimental Accuracy
DegreeSAD	0.978	0.746
CoDetector	0.7143	0.78573
BayesDetector	0.928	0.923
SemiSAD	0.936	0.931
PCASelectUsers	0.928	0.466005
FAP	0.558	0.39375

Based on the comparison between our experimental results and the results proposed by the original authors, using the same dataset, features, filler size, and attack size, we observed that our accuracy closely aligns with the authors' results, with one exception. Specifically, the PCASelectUsers, FAP and Degree SAD metric yielded smaller results than the original findings. This disparity can be attributed to the authors' utilization of different feature-extraction techniques.

VI. CONCLUSION

In this study, we applied our model to two different datasets: Amazon data and movie lens. The Amazon dataset consists of 3921 users, 14711 items, and 10368 records, whereas the Movie Lens dataset has 1337 users, 2068 records, and 37042 ratings. Our aim is to detect shilling attacks in recommender systems using various detection techniques.

First, we discuss different types of shilling attacks and provide a brief description of each. We then analyzed how some obfuscated attack models were derived from standard attacks. We also examined and categorized the characteristic traits used in the detection process. In addition, we explored different detection and robust algorithms available for shilling attack detection.

To evaluate the performance of the attack detection models, we used metrics such as accuracy, F1-measure, recall, precision, macro-average, weighted average, and execution times. We conducted experiments on two benchmark rating datasets and compared their results. Specifically, we focused on commonly utilized shilling detection strategies and compared our findings with the theoretical proposals of the original authors.

The experimental results showed that the hybrid deep learning algorithms achieved the best performance, followed by supervised and semi-supervised learning algorithms. However, the unsupervised method performs poorly.

We also observed that hyper deep learning-based Shilling Attack Detection demonstrated accuracy and quality in identifying various mixed attacks, even with a smaller number of datasets for evaluation. For future improvements, employing larger datasets with varying levels of sparsity would enhance the effectiveness of these models. In addition, a comparison between hybrid deep learning algorithms and hybrid supervised learning algorithms can be conducted. Moreover, we obtained promising results in terms of detection rate, accuracy, precision, recall, and F-measure by conducting experiments on Movie Lens datasets of various sizes and sparsity.

Moving forward, our research focuses on exploring attack possibilities and detection methods for multi-criteria collaborative filtering. We aimed to investigate detection strategies that can effectively detect shilling attacks in scenarios involving multiple criteria.

ACKNOWLEDGMENT

This work was supported and funded by Cairo University, Egypt, and Midocean University, Comoros. The authors, therefore, acknowledge with thanks them for technical and financial support.

REFERENCES

[1] R. A. Zayed, L. F. Ibrahim, H. A. Hefny, and H. A. Salman, "Shilling attacks detection in collaborative recommender system: Challenges and promise," in *Proc. Workshops Int. Conf. Adv. Inf. Netw. Appl.*, Italy, 2020, pp. 429–439.

[2] Y. Hao, F. Zhang, J. Wang, Q. Zhao, and J. Cao, "Detecting shilling attacks with automatic features from multiple views," *Secur. Commun. Netw.*, vol. 2019, pp. 1–13, Aug. 2019.

[3] L. Jiang, Y. Cheng, L. Yang, J. Li, H. Yan, and X. Wang, "A trust-based collaborative filtering algorithm for e-commerce recommendation system," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 8, pp. 3023–3034, Aug. 2019.

[4] Z. Batmaz, B. Yilmazel, and C. Kaleli, "Shilling attack detection in binary data: A classification approach," *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 6, pp. 2601–2611, Jun. 2020.

[5] S. Alonso, J. Bobadilla, F. Ortega, and R. Moya, "Robust model-based reliability approach to tackle shilling attacks in collaborative filtering recommender systems," *IEEE Access*, vol. 7, pp. 41782–41798, 2019.

[6] K. Chen, P. P. K. Chan, F. Zhang, and Q. Li, "Shilling attack based on item popularity and rated item correlation against collaborative filtering," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 7, pp. 1833–1845, Jul. 2019.

[7] W. Zhou, J. Wen, Y. S. Koh, Q. Xiong, M. Gao, G. Dobbie, and S. Alam, "Shilling attacks detection in recommender systems based on target item analysis," *PLoS ONE*, vol. 10, no. 7, Jul. 2015, Art. no. e0130968.

[8] B. Mobasher, R. Burke, R. Bhaumik, and J. J. Sandvig, "Attacks and remedies in collaborative recommendation," *IEEE Intell. Syst.*, vol. 22, no. 3, pp. 56–63, May 2007.

[9] A. P. Sundar, F. Li, X. Zou, T. Gao, and E. D. Russomanno, "Understanding shilling attacks and their detection traits: A comprehensive survey," *IEEE Access*, vol. 8, pp. 171703–171715, 2020.

[10] Y. Hao, P. Zhang, and F. Zhang, "Multiview ensemble method for detecting shilling attacks in collaborative recommender systems," *Secur. Commun. Netw.*, vol. 2018, pp. 1–33, Oct. 2018.

[11] W. Zhou, "Abnormal group user detection in recommender systems using multi-dimension time series," in *Proc. Collaborate Comput., Netw., Appl. Worksharing, 12th Int. Conf., CollaborateCom*, Beijing, China, 2016, pp. 373–383.

[12] P.-A. Chirita, W. Nejdl, and C. Zamfir, "Preventing shilling attacks in online recommender systems," in *Proc. 7th Annu. ACM Int. Workshop Web Inf. Data Manage.*, Nov. 2005, pp. 1–8.

[13] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik, "Classification features for attack detection in collaborative recommender systems," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2006, pp. 542–547.

[14] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik, "Detecting profile injection attacks in collaborative recommender systems," in *Proc. 8th IEEE Int. Conf. E-Commerce Technol. 3rd IEEE Int. Conf. Enterprise Comput., E-Commerce, E-Services (CEC/EEE)*, Jun. 2006, p. 23.

[15] C. A. Williams, B. Mobasher, and R. Burke, "Defending recommender systems: Detection of profile injection attacks," *Service Oriented Comput. Appl.*, vol. 1, no. 3, pp. 157–170, Oct. 2007.

[16] F. Zhang and Q. Zhou, "A meta-learning-based approach for detecting profile injection attacks in collaborative recommender systems," *J. Comput.*, vol. 7, no. 1, pp. 226–234, Jan. 2012.

[17] W. Zhou, J. Wen, Q. Xiong, M. Gao, and J. Zeng, ''SVM-TIA a shilling attack detection method based on SVM and target item analysis in recommender systems,'' *Neurocomputing*, vol. 210, pp. 197–205, Oct. 2016.

[18] Z. Yang, L. Xu, Z. Cai, and Z. Xu, ''Re-scale AdaBoost for attack detection in collaborative filtering recommender systems,'' *Knowl.-Based Syst.*, vol. 100, pp. 74–88, May 2016.

[19] Z. Huang, X. Xu, H. Zhu, and M. Zhou, ''An efficient group recommendation model with multiattention-based neural networks,'' *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4461–4474, Nov. 2020.

[20] R. A. Zayed, H. A. Hefny, L. F. Ibrahim, and H. A. Salman, ''An enhanced method for detecting attack in collaborative recommender system,'' in *Proc. 1st Int. Conf. Adv. Innov. Smart Cities (ICAISC)*, Jan. 2023, pp. 1–5.

[21] K. Bryan, M. O'Mahony, and P. Cunningham, ''Unsupervised retrieval of attack profiles in collaborative recommender systems,'' in *Proc. ACM Conf. Recommender Syst.*, Oct. 2008, pp. 1–13.

[22] B. Mehta, T. Hofmann, and P. Fankhauser, ''Lies and propaganda: Detecting spam users in collaborative filtering,'' in *Proc. 12th Int. Conf. Intell. User Interfaces*, Jan. 2007, pp. 1–8.

[23] R. Bhaumik, B. Mobasher, and R. Burke, ''A clustering approach to unsupervised attack detection in collaborative recommender systems,'' in *Proc. Int. Conf. Data Sci. (ICDATA)*, 2011, pp. 1–7.

[24] C.-Y. Chung, P.-Y. Hsu, and S.-H. Huang, ''A novel approach to filter out malicious rating profiles from recommender systems,'' *Decis. Support Syst.*, vol. 55, no. 1, pp. 314–325, 2013.

[25] A. Bilge, Z. Ozdemir, and H. Polat, ''A novel shilling attack detection method,'' *Proc. Comput. Sci.*, vol. 31, pp. 165–174, Jan. 2014.

[26] Z. Yang, Z. Cai, and Y. Yang, ''Spotting anomalous ratings for rating systems by analyzing target users and items,'' *Neurocomputing*, vol. 240, pp. 25–46, May 2017.

[27] C. Panagiotakis, H. Papadakis, and P. Fragopoulou, ''Unsupervised and supervised methods for the detection of hurriedly created profiles in recommender systems,'' *Int. J. Mach. Learn. Cybern.*, vol. 11, no. 9, pp. 21–2165, 2020.

[28] Z. Xiaojin, ''Semi-supervised learning literature survey,'' Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, Tech. Rep., TR1530, 2008.

[29] Z. Wu, J. Wu, J. Cao, and D. Tao, ''HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation,'' in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2012, pp. 1–9.

[30] C. Tong, X. Yin, J. Li, T. Zhu, R. Lv, L. Sun, and J. J. P. C. Rodrigues, ''A shilling attack detector based on convolutional neural network for collaborative recommender system in social aware network,'' *Comput. J.*, vol. 61, no. 7, pp. 949–958, Jul. 2018.

[31] Q. Zhou, J. Wu, and L. Duan, ''Recommendation attack detection based on deep learning,'' *J. Inf. Secur. Appl.*, vol. 52, Jun. 2020, Art. no. 102493.

[32] S. Zhang, A. Chakrabarti, J. Ford, and F. Makedon, ''Attack detection in time series for recommender systems,'' in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2006, pp. 1–6.

[33] N. Hurley, Z. Cheng, and M. Zhang, ''Statistical attack detection,'' in *Proc. 3rd ACM Conf. Recommender Syst.*, Oct. 2009, pp. 149–156.

[34] B. Mehta and W. Nejdl, ''Unsupervised strategies for shilling detection and robust collaborative filtering,'' *User Model. User-Adapted Interact.*, vol. 19, pp. 65–97, Jul. 2009.

[35] R. Bhaumik, ''Securing collaborative filtering against malicious attacks through anomaly detection,'' in *Proc. 4th Workshop Intell. Techn. Web Personalization (ITWP)*, Boston, MA, USA, 2006, p. 10.

[36] J. Cao, Z. Wu, B. Mao, and Y. Zhang, ''Shilling attack detection utilizing semi-supervised learning method for collaborative recommender system,'' *World Wide Web*, vol. 16, nos. 5–6, pp. 729–748, 2013.

[37] J. A. Nasir, O. S. Khan, and I. Varlamis, ''Fake news detection: A hybrid CNN-RNN based deep learning approach,'' *Int. J. Inf. Manage. Data Insights*, vol. 1, no. 1, Apr. 2021, Art. no. 100007.

[38] K. Vivekanandan and N. Praveena, ''Hybrid convolutional neural network (CNN) and long-short term memory (LSTM) based deep learning model for detecting shilling attack in the social-aware network,'' *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 1, pp. 1197–1210, Jan. 2021.

[39] E. Mahsa and R. Kashef, ''Detecting shilling attacks using hybrid deep learning models,'' *Symmetry*, vol. 12, no. 11, p. 1805, 2020.

[40] M. Aktukmak, Y. Yilmaz, and I. Uysal, ''Quick and accurate attack detection in recommender systems through user attributes,'' in *Proc. 13th ACM Conf. Recommender Syst.*, Sep. 2019, pp. 348–352.

[41] S. Zhang, H. Yin, T. Chen, Z. Huang, L. Cui, and X. Zhang, ''Graph embedding for recommendation against attribute inference attacks,'' in *Proc. Web Conf.*, Apr. 2021, pp. 3002–3014.

[42] W. Zhou, J. Wen, Q. Qu, J. Zeng, and T. Cheng, ''Shilling attack detection for recommender systems based on credibility of group users and rating time series,'' *PLoS ONE*, vol. 13, no. 5, May 2018, Art. no. e0196533.

[43] N. Praveena and K. Vivekanandan, ''A survey on detection approaches of shilling attacks in SAN,'' in *Proc. 5th Int. Conf. Comput. Methodologies Commun. (ICCMC)*, Apr. 2021, pp. 233–238.

[44] J. Wang, ''Ada: Adversarial learning based data augmentation for malicious users detection,'' *Appl. Soft Comput.*, vol. 117, Mar. 2022, Art. no. 108414.

[45] M. Si and Q. Li, ''Shilling attacks against collaborative recommender systems: A review,'' *Artif. Intell. Rev.*, vol. 53, no. 1, pp. 291–319, Jan. 2020.

[46] Z. Chen and S. Wang, ''A review on matrix completion for recommender systems,'' *Knowl. Inf. Syst.*, vol. 64, pp. 1-34, Jan. 2022.

[47] T. Dou, ''Collaborative shilling detection bridging factorization and user embedding,'' in *Proc. Collaborative Comput., Netw., Appl. Worksharing, 13th Int. Conf., CollaborateCom*, Edinburgh, U.K., 2017, pp. 459–469.

[48] M. Ebrahimian and R. Kashef, ''A CNN-based hybrid model and architecture for shilling attack detection,'' in *Proc. IEEE Can. Conf. Electr. Comput. Eng. (CCECE)*, Sep. 2021, pp. 1–7.

[49] C. Krügel, T. Toth, and E. Kirda, ''Service specific anomaly detection for network intrusion detection,'' in *Proc. ACM Symp. Appl. Comput.*, Mar. 2002, pp. 1–9.

REDA A. ZAYED received the M.Sc. degree in computer science from the Faculty of Graduate Studies for Statistical Research, Cairo University, in 2017, where he is currently pursuing the Ph.D. degree in computer science. He is the Head of the Software Development Department, Ministry of Justice, Egypt, and the College of Informatics, Midocean University. He has more than 16 years of experience involved in product management and data science. His research interests include artificial intelligence, machine learning, advanced database management, knowledge-based systems, big data, and data science.

LAMIAA FATTOUH IBRAHIM received the B.Sc. degree from the Department of Computer and Automatic Control, Faculty of Engineering, Ain Shams University, in 1984, the master's degree from École National Supérieur de Télécommunication (ENST), Paris, in 1987, the master's degree from the Department of Computer and Systems Engineering, Faculty of Engineering, Ain Shams University, in 1993, and the Ph.D. degree from the Faculty of Engineering, Cairo University, in 1999, Previously, she was with the Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University. She was also the Vice Dean of Education and Student Affairs with the Faculty of Information Systems and Computer Science, October 6 University, and the Head of the Department of Computer Science, Faculty of Graduate Studies for Statistical Research (FGSSR), Cairo University, where she is currently a Professor of artificial intelligence and the Dean of College of Informatics, Everyone's Smart University. She has more than 39 years of experience in the fields of network design engineering and artificial intelligence, focusing on applying knowledge base and data mining techniques to wired and wireless network planning. She has published papers in many international journals and international conferences in the areas of networks, data mining, and wired and mobile network planning.

HESHAM A. HEFNY received the B.Sc., M.Sc., and Ph.D. degrees in electronics and communication engineering from Cairo University, in 1987, 1991, and 1998 respectively. He is currently a Professor of computer science with the Faculty of Graduate Studies for Statistical Research (FGSSR), Cairo University, where he is also the Vice Dean of Graduate Studies and Research. He has authored more than 160 papers in international conferences, journals, and book chapters. His major research interests include computational intelligence (neural networks–fuzzy systems-genetic algorithms–swarm intelligence), data mining, and uncertain decision-making. He is a member of the following professional societies, such as IEEE Computer, IEEE Computational Intelligence, and IEEE System, Man, and Cybernetics.

HESHAM A. SALMAN received the master's degree in engineering from Ain Shams University, in 1996, and the Ph.D. degree from the Faculty of Computing and Information Systems, Ain Shams University. He is currently with the College of Informatics, Midocean University, and the Higher Institute of Computer and Information Technology, Alshrouk Academy, Cairo, Egypt. Previously, he was with the Department of Information System, Faculty of Computing and Information Technology , King Abdulaziz University. He was also the General Manager of the Technology Competency Center, Ministry of Commerce, and Industry, Cairo. He has participated in many studies and works related to information systems. He has more than 37 years of experience in the fields of network engineering, computer security, and programming applications. He has published in networks, data mining, and wire and mobile network planning.

ABDULAZIZ ALMOHIMEED received the Ph.D. degree from the University of Southampton, England, and the master's degree from Monash University, Australia. He is currently an Assistant Professor with the College of Computer and Information Sciences, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia. He is passionate about leveraging technology to create innovative solutions. His current research interests include natural language processing, artificial intelligence, data science, the Internet of Things, and network security.